A cognitive information processing framework for distributed sensor networks

Feiyi Wang[a] and Hairong Qi[b]

[a]MCNC Research & Development Institute, Research Triangle Park, NC, USA
[b]Hairong Qi, Electrical and Computer Engineering, University of Tennessee, TN, USA

ABSTRACT

In this paper, we present a cognitive agent framework (CAF) based on swarm intelligence and self-organization principles, and demonstrate it through collaborative processing for target classification in sensor networks. The framework involves integrated designs to provide both *cognitive behavior* at the organization level to conquer complexity and *reactive behavior* at the individual agent level to retain simplicity. The design tackles various problems in the current information processing systems, including overly complex systems, maintenance difficulties, increasing vulnerability to attack, lack of capability to tolerate faults, and inability to identify and cope with low-frequency patterns. An important and distinguishing point of the presented work from classical AI research is that the acquired intelligence does not pertain to distinct individuals but to groups. It also deviates from multi-agent systems (MAS) due to sheer quantity of extremely simple agents we are able to accommodate, to the degree that some loss of coordination messages and behavior of faulty/compromised agents will not affect the collective decision made by the group.

1. INTRODUCTION

The rapid advances and infusion of the Micro Electro Mechanical System (MEMS) and wireless communication technologies have fostered the development of sensor networks. A large amount of low cost, intelligent microsensors can be rapidly deployed in an environment of interest. These sensors can individually sense the environment, or collaborate with each other and achieve complex information gathering and dissemination tasks like environment monitoring, intrusion detection, target tracking, remote sensing, surveillance, etc. These tasks are usually time-critical and require reliable delivery of accurate information.

Although potentially powerful, several unique characteristics of sensor networks present interesting design challenges to the information processing system. We summarize the unique characteristics of sensor networks from four aspects: scalability, reliability, dynamics, and stringent resource.

Scalability. The proliferation of low cost sensors enables large amounts of sensor deployment. As more sensors are put into the field, more data will be captured, which could potentially enhance decision making. However, the risk of large data transfer and information overloading is also greatly increased.

Reliability. Sensors communicate through low-bandwidth and unreliable wireless links compared to wired communication. An individual sensor may suffer intermittent connectivity due to high bit error rate (BER) of the wireless link, and it can be further deteriorated by environmental hazard. The challenge is to provide reliable information based on potentially unreliable wireless communication networks and unreliable sensor nodes.

Dynamics. Since sensors are usually rapidly deployed in large amount, it is very difficult, if not impossible, to maintain a pre-designed network structure. Sensors may be static or mobile. They may come and go because of new sensor deployment and node failure. All these dynamic features indicate that sensor networks tend to be infrastructureless and require the underlying network services and applications to be adaptive.

Stringent resource. Sensor nodes generally consist of four basic units[1]: sensing, processing, communication, and power. The lifetime of a sensor node is mainly determined by the power supply since battery replacement is not an option. The longer the sensors last, the more stable the network, and the less dynamic the network.

Further author information: send correspondence to Feiyi Wang, E-mail: `fwang2@anr.mcnc.org`

In order to save power, redundant activities should be reduced if not eliminated. For example, communications should be constrained since it has been shown[2, 3] to be the most power-consuming process.

All the above characteristics have profound implications in the design of information processing system in sensor networks. For example, redundancy is desired in order to perform robust and fault-tolerant decision making, especially when individual sensor nodes are potentially unreliable. The stringent resource issue also indicates low bandwidth supply and unreliable wireless links. How to perform information processing for complex data dissemination tasks within the limit of the wireless communication networks can be a challenging problem. On the other hand, the size of the microsensor limits the computational power of individual sensor node such that complex algorithms are not suitable to be performed in the context of sensor networks.

We present a cognitive agent framework (CAF) to support collaborative information processing in sensor networks. The framework involves integrated designs to provide both *cognitive behavior* at the organization level to conquer complexity and *reactive behavior* at the individual agent level to retain simplicity. Within this framework, we explore three issues, including (1) lightweight and fault tolerant infosphere as environment for multiple interaction; (2) applicability of fundamental concepts of self-organization principles, which include positive feedback for amplification of correct or desirable behavior; negative feedback for decay to counterbalance and stabilize the collective pattern; and random walks for new solutions and findings; and (3) role-switching and recruiter-driven architecture that exhibits both reactive behaviors at the individual agent level to retain simplicity and cognitive behavior at the organization level. We will demonstrate the effectiveness of CAF through a real-world problem solving example for collaborative target classification.

2. COGNITIVE AGENT FRAMEWORK (CAF)

We classify artificial agent into two broad categories: cognitive and reactive. *Cognitive agents* are said to be "intentional", that is, it has explicitly defined goals and subsequent plans to achieve these goals. *Reactive agents*, on the other hand, react based on the stimuli from limited view of environment they are connected with. Therefore, their behaviors are triggered by a combination of internal drives and external stimulations. Pure cognitive agents should have symbolic representation of the environment (mental map) based on what they can reason with, while reactive agents lack that representation. However, we believe that both types of agents are capable of accomplishing complex tasks and the line of differentiation is blurred, especially with the consideration of advanced development of large quantities of computing elements, such as sensors.

Our proposed CAF architecture design exhibits both reactive behaviors (as seen from the perspective of a single agent) and cognitive behaviors (as seen from the perspective of swarm organization as a whole). The technical rationale is organized around three central topics: First, we discuss design issues on constructing single *stimuli*-driven swarm agent[4–6] and how these agents interact with each other through *shared space*. The primary characteristics exposed at this level are mostly reactive. Second, we present a novel *role-switching* architecture for organizing the agents to efficiently carry out specific tasks. The general goal for the system as a whole is to show collective intelligence, adaptability and fault tolerance capability to a variety of changes and uncertainty without compromising the defined objectives. Finally, we demonstrate swarm intelligence in distributed sensor network context and show the potential gains by utilizing the proposed CAF.

2.1. *Stimuli*-Driven Swarm Agent (SA)

The design theme of *stimuli*-driven agent is its extreme simplicity. However, simplicity is not synonymous with incapacity. Instead, each agent is designed to be a live organic element that has limited perception of the environment, simple reasoning or deliberation process, and ability to take actions. It influences the behavior of the organization by modifying the environment through both positive and negative feedback, which are the underpinning elements for self-organization. To facilitate the discussion, we make the following definitions: the objective is realized by a sequence of tasks: $Goal : \langle T_1, T_2, \cdots, T_n \rangle$; each task T_i is associated with three parameters, *weight*, *stimuli*, and *constraints*, which indicate the priority of the task from goal planner point of view, the attractiveness to the task from agent point of view, and the satisfaction constraints for the task, respectively. We denote these parameters as a tuple $T_i : \langle w_i, s_i, c_i \rangle$, which will be communicated through the environment (or the infosphere) where multiple agents interact.

Multiple Interactions: A fundamental underlying support for multiple interactions is what we call a lightweight infosphere for indirect communication. Action and interaction are considered the motor elements to enable emergent functionality, one of the founding processes for self-organization. Reasoning is done locally and acting is achieved through modifying the environment, i.e., the infosphere, denoted as Ω . SAs are endowed with autonomy in the sense that they will not be preprogrammed to reach decision or take commands from central controller, instead they are directed by a set of tendencies, which in our work, is the satisfaction functions. We envision a range of operations can be defined on this infosphere Ω, such as `read`, `takeIfExist`, `readIfExist`, etc., which are the primary means for SAs to interact with each other and make impacts to the overall environment.

Perception: In CAF, agents only have a partial representation of the environment. In other words, the perception of single agent is local. We consider perception as a function, which is associated with a set of values called *percepts* with the set of state variables Σ in the infosphere following the algebra put forward by Genesereth and Nilsson.[7] If we let P_A be the set of percepts associated with agent A, then perception function can be defined as $Percept(A) : \Sigma \longrightarrow P_A$.

Deliberation and Execution: The SA deliberation process will take into account the following factors: importance of the task, relevance/attractiveness of the task, and local conditions (e.g., energy saving or self survivals, etc.). The goal is to select a task to maximize its own net utility gains. The objective function can take the general form such as

$$Maximize\{T_i = \frac{w_i}{\sum_{i=1}^{n} w_i} s_i L_a\}$$

where i indicates a specific task, a indicates an agent, and L_a is the local condition. When task i is selected by an agent, its weight can be modified to reflect positive feedback: one simple example is to make $w_i(t+1) = w_i(t)+1$, which will show a gradual effect of attracting more agents working on the same task. By the same token, we might also want to apply certain constraints for negative feedback to prevent the scenario where an excessive number of agents are all working on the same task.

Generally speaking, an agent's action is highly task specific. Our design idea is to incorporate the concept of "fading with time": an agent will write its task result into the infosphere, thus making it visible to all other agent performing the same task. This result will fade away with time unless other agents strengthen it by giving the same or similar observation. A task is regarded as complete when both the constraint c_i is met and agents performing the task reach relative consensus. Obviously, a threshold activation function is needed to denote the degree of consensus.

2.2. Role-switching, Recruiter-driven Swarm Agent Organization

Our discussion has thus far focused on the micro level, i.e., design of individual agent. High-level properties such as goal performing efficiency, adaptability and fault tolerance can only be achieved through effective and flexible agent organization. The agents should organize based on desirable emergent behavior from a rule base. There needs to be a tension between rules for fluidity of self-organization and efficiency rules. The swarm needs to quickly converge to a steady state based on current conditions, dampening minor conditions for efficient operation, while allowing major changes to reorganize the swarm.

The common methods for agent organization are self-organized peer-to-peer and centralized hierarchy model. The former carries the potential to exhibit emergent behavior therefore the ability to adapt to changes and cope with unplanned events, but at times its behaviors can be unpredictable and uncontrollable. The later can have great efficiency and predictability only when structure elements and environment are relatively static and everything is in control. CAF harmonizes the conflicting factors by allowing agent to switch roles on demand. As illustrated in Fig. 1, two types of agents are defined, recruiters and workers, organized by relative specialization, from more cognitive to more reactive agents. The recruiting agents carry high-level goal setting, use *stimuli* to attract other agents to work and monitor task results in the swarm. They emerge via distributed elections, with the "oldest" capable agent in the set being picked for the job. Once dispatched or elected to be a recruiter, they are responsible for resource discovery, task dispatch and monitoring, and task adjustment (e.g., by fine tuning $\langle w_i, s_i, c_i \rangle$).

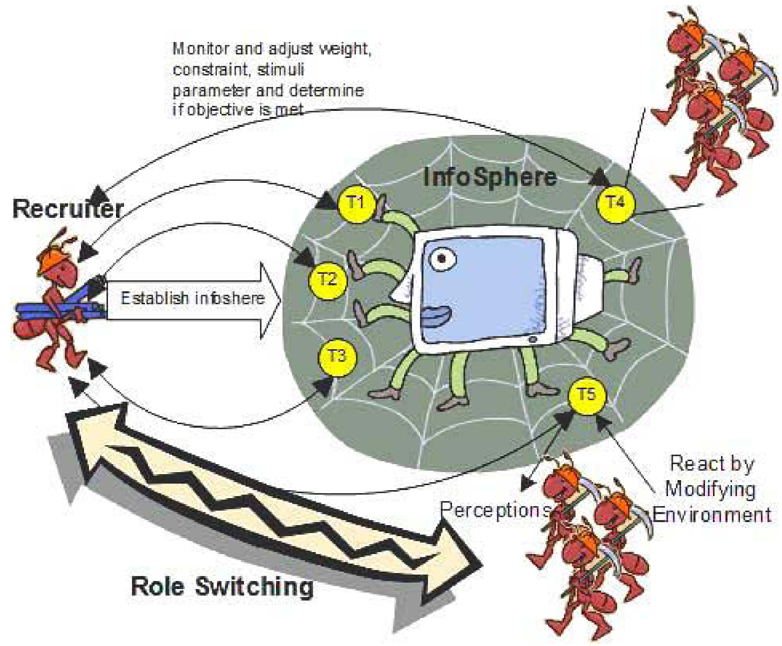


Figure 1. The cognitive agent framework (CAF).

2.3. Collaborative Intelligence in Sensor Networks

In a sensor network, a large amount of low cost, intelligent microsensors can be rapidly deployed in an environment of interest. These sensors can individually sense the environment. They can also collaborate with each other and achieve complex information gathering and dissemination tasks. Since individual sensors can only sense a portion of the sensor field using certain sensing modalities, information provided by single sensor might as well be biased or inaccurate. If the sensor is malfunctioning or even tempered by malicious adversaries, the information can be contradictory. Therefore, collaborative intelligence among multiple sensor nodes is important to complement for each other's missing information and tolerate faults.

Sensor networks form a typical distributed environment, where various computing paradigms can be used to achieve collaborative information processing. The most popular computing models deployed are illustrated in Fig. 2. Fig. 2(a) shows one extreme case of collaborative processing where all sensor nodes send the raw data to a processing center for data dissemination. This computer model is easy to implement but suffers from large data transfer and slow response. In order to avoid transferring the raw data, sensor nodes can perform some local processing and send a compressed version of the raw data or simply the local processing results to the processing center, as illustrated in Fig. 2 (b). This scheme is widely used in distributed detection,[8] where each sensor node only transfers the local detection result derived from local processing. In other applications, collaborative processing can be carried out hierarchically with multiple levels of processing centers in order to solve scalability problems.[9] This model is illustrated in Fig. 2 (c).

From the above discussion, we observe that collaborative processing can be done at different levels, the raw data level, the decision level, and some intermediate levels. We have then come to realize that collaborative information processing should be both an intrinsic component of sensor network and one of the fundamental means to achieve accurate decision making. However, most collaborative processing algorithms developed so far are in general *reactive* or event-driven, which then limits the sensor network's capability in reasoning, learning, and self-organizing. Using CAF in this research effort, the sensor network is able to derive *intelligence*. We apply this framework in a specific application - target classification with intrusion tolerance,[10–12] and evaluate the advantages of using swarm intelligence.

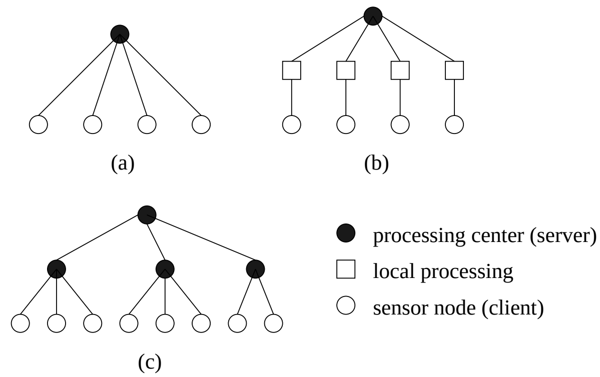
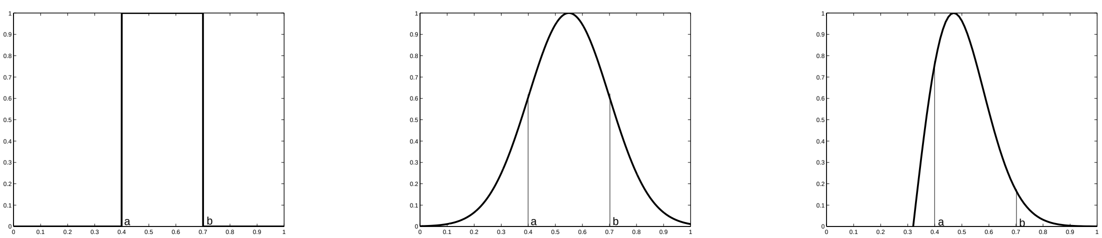


Figure 2. Various computing models for collaborative processing.



(a) Uniform (b) Gaussian (c) Rayleigh

Figure 3. Possible choices for modelling the distribution of the confidence within certain confidence range $[a, b] = [0.4, 0.7]$.

3. CAF FOR COLLABORATIVE INTRUSION TOLERANCE

There are three enabling components for collaborative intelligence using CAF, including local knowledge representation, integration of local knowledge at the infosphere, and fault tolerance and consensus state. We will address each component in the following three subsections.

3.1. Local Knowledge Representation

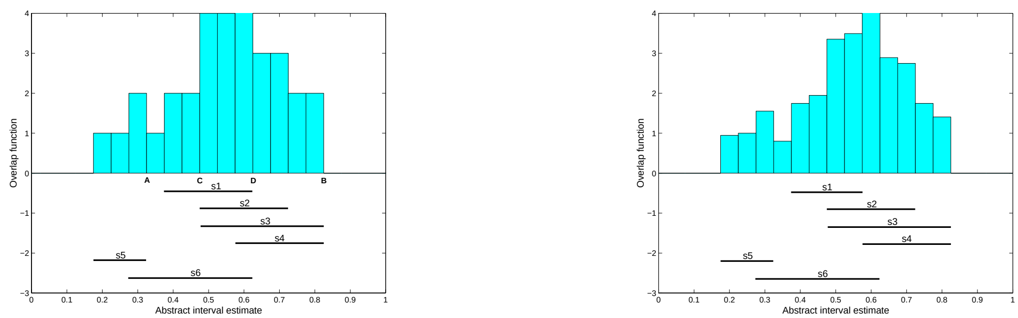
An important concept in the derivation of the fault-tolerant algorithm is the *interval*-based collaboration as compared to the *value*-based collaboration. The value-based collaboration performs the coordination algorithm based on inputs from sensors which, without loss of generality, we assume to be concrete numbers (be it integers or floating point numbers). The interval-based collaboration performs the coordination algorithm based on an interval clustered around the physical readout (or the number).

We assume each sensor behave as an *abstract device*. We define an abstract device as a device that reads a physical parameter and gives out an abstract interval estimate which is a bounded and connected subset of the real number. Based on this definition, a *correct device* is an abstract device whose interval estimate contains the actual value of the parameter being measured. Otherwise, it is a *faulty device*. A faulty device is *tamely faulty* if its interval estimate overlaps with a correct device, and is *widely faulty* if it does not overlap with any correct device. For example, in the application of classifying a certain target, the sensor output might be expressed as "I am 40 to 70 percent sure that the target just went by me is a diesel truck." We define this range as the confidence interval estimate. The confidence itself can be modelled by different stochastic distributions, the simplest of which would be a uniform distribution, where equal weight has been registered on each confidence within the confidence range. Other appropriate distributions could be a Gaussian (more weight on the central confidence) or a Rayleigh (more weight on the low confidence) as shown in Fig. 3.

Using the $\langle w_i, s_i, c_i \rangle$ representation that each agent generates associated the task of target classification, w_i is the weight distribution function illustrated above, s_i is the confidence range $[a_i, b_i]$ that the agent has over the local decision, and c_i is the required accuracy rate specified by the user. For example, the user will only accept classification result with an accuracy rate higher than 70%. Upon generating local knowledge in the format of $\langle w_i, s_i, c_i \rangle$, each agent will upload this information into the infospere. Newly uploaded information will be integrated with existing ones, thus deriving collaborative intelligence.

3.2. Integration of Local Knowledge at the Infosphere

In order to provide collaborative intelligence out of local information derived, we modify the original multi-resolution integration (MRI) algorithm developed by Prasad, Iyengar, and Rao in 1994,[13] where a processing center collects the outputs of the sensors and constructs an overlap function. Two examples of the overlap function using a uniform weight function [Fig. 4(left)] and a Gaussian weight function [Fig. 4(right)] are shown in Fig. 4.



(a) Uniform (b) Gaussian

Figure 4. The overlap function for a set of 6 sensors using two distribution models (resolution is 0.05).

An integrated solution will be derived at the infosphere from the overlap function as soon as any agent uploads its local information. We choose to return the interval with the overlap function ranges $[n - f, n]$ where f is the number of faulty sensors and n is the total number of sensors in the network. This algorithm satisfies a Lipschitz condition which ensures that minor changes in the input intervals cause only minor changes in the integrated result. In addition, the algorithm is able to reduce the width of the output interval in most cases and produce a narrower output interval when the number of sensors involved is large.

3.3. Fault Tolerance and Consensus State

In the above-mentioned integration algorithm, based on different number of faulty sensors, we can derive different versions of integration interval. The upper bound of f can be derived from the Byzantine Generals problem, where the maximum number of faults (f) that certain amount of sensor nodes (n) can tolerate is

$$f = \lfloor \frac{n-1}{3} \rfloor.$$

The advantage of using the overlap function in the infospere is that a partially integrated result can always be derived in response to each agent's local reaction. If we choose the centrum of the integrated interval as a measure of the accuracy rate and if this rate meets what's required by the user (c_i), then the task has been accomplished, that is, a consensus has been reached at this shared space.

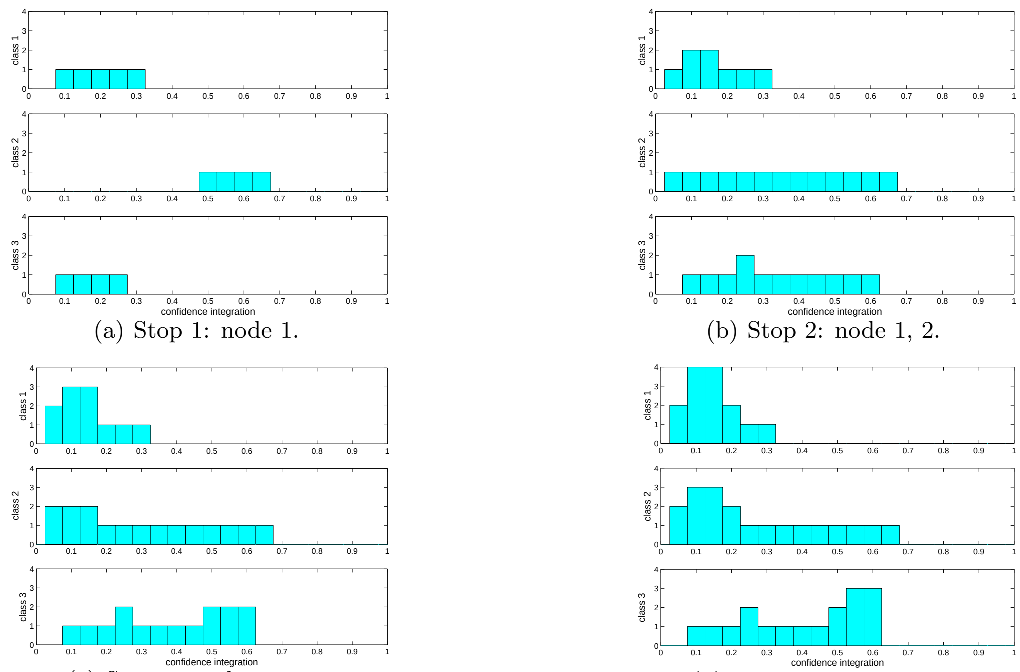
We present a case study that uses collaborative target classification to show how intrusion tolerance using fusion can help ensure the correctness of the final decision. The collaborative processing is carried out among a cluster of 4 sensor nodes to determine on three possible targets as shown in Table 1. The three intervals associated with each node indicate the classification confidence range of that node *thinking* the target might be of class 1, 2, or 3. For example, node 1 classifies the target as class 1 with a confidence ranging from 0.10 to 0.29. In this example, the ground truth shows the target should be class 3. We assume node 1 is tampered and thus provides a faulty result.

Figure 5 illustrates how the partially integrated confidence range is generated when the fusion process is carried out from node 1 to node 4 in sequence. We assume the resolution requirement is 0.05, then the size of the

nodes	class 1	class 2	class 3
1	[0.10, 0.29]	[0.46, 0.65]	[0.10, 0.21]
2	[0.05, 0.14]	[0.05, 0.41]	[0.22, 0.58]
3	[0.05, 0.15]	[0.05, 0.15]	[0.49, 0.59]
4	[0.08, 0.16]	[0.08, 0.16]	[0.51, 0.60]

Table 1. Local classification confidence range.

1-D array is 21*3=63 to hold results for the three targets. The four sub-figures show the content of this array while the fusion process progresses. Table 2 summarizes the decision making procedure at each sensor node. We observe that the fusion result at nodes 1 and 2 is "class 2" but changes to "class 3" at nodes 3 and 4.



(a) Stop 1: node 1.

(b) Stop 2: node 1, 2.

(c) Stop 3: node 1, 2, 3.

(d) Stop 4: node 1,2,3,4.

Figure 5. Multi-sensor collaboration result at each sensor node.

	stop 1		stop 2		stop 3		stop 4	
	c	acc	c	acc	c	acc	c	acc
class 1	1	0.2	0.5	0.125	0.75	0.125	1	0.125
class 2	**2.3**	**0.575**	**4.55**	**0.35**	0.6	0.1	0.75	0.125
class 3	0.7	0.175	0.5	0.25	**3.3**	**0.55**	**3.45**	**0.575**

Table 2. Decision making procedure at each node.

4. CONCLUSION

The proposed cognitive agent framework (CAF) presented the swarm agent design which can perceive, deliberate, and act based on self-organization principles. Although from the perspective of individuals, a swarm agent exhibits mostly reactive (stimuli-driven) behaviors, the organization framework we presented united the system as a whole to possess properties of robustness, adaptability, scalability and manageability. Furthermore, the proposed collaborative information processing in distributed sensor network by utilizing the swarm agent framework brings us one step closer to real world problem solving.

ACKNOWLEDGMENTS

The work was supported in part by DARPA under contract N66001-001-8946 and in part by DARPA under contract N66001-00-C-8057 from the Space and Naval Warfare Systems Center - San Diego (SPAWARSYSCEN). The views, opinions and findings contained in this paper are those of the authors and should not be construed as official DARPA or SPAWARSYSCEN's positions, policy or decision.

REFERENCES

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks* **38**, pp. 393–422, 2002.
2. S. Meguerdichian, S. Slijepcevic, V. Karayan, and M. Potkonjak, "Localized algorithms in wireless ad-hoc networks: location discovery and sensor exposure," in *MOBIHOC*, 2001.
3. V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine* , pp. 40–50, March 2002.
4. E. Bonabeau, M. Dorigo, and G. Thraulaz, *Swarm intelligence: from natural to artificial systems*, Oxford University Press, 1999.
5. E. Bonabeau and G. Thraulaz, "Swarm smarts," *Scientific American* , pp. 72–79, March, 2000.
6. J. Kennedy and R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001.
7. M. R. Genesereth and N. Nilsson, *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, 1987.
8. R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors: Part I - Fundamentals," *Proc. IEEE* **85**, pp. 54–63, January 1997.
9. "Ivy - A sensor network infrastructure for the college of engineering." `http://www-bsac.eecs.berkeley.edu/projects/ivy/`.
10. S. Bryant and F. Wang, "Aspects of adaptive reconfiguration in building intrusion tolerant server systems," *(to appear) Journal of Complexity* , 2004.
11. F. Wang, R. Upalli, and C. Killian, "Analysis of intrusion tolerant techniques for distributed server systems," in *Military Communication Conference (MILCOM)*, (Boston, MA), Oct 13 - 15 2003.
12. R. Wang, F. Wang, and G. T. Byrd, "Design and implementation of acceptance monitor for building intrusion tolerant systems," *Journal of Software - Practice and Experience* (33), pp. 1399–1417, 2003.
13. L. Prasad, S. S. Iyengar, and R. L. Rao, "Fault-tolerant sensor integration using multiresolution decomposition," *Physical Review E* **49**, pp. 3452–3461, April 1994.